AD NUMBER

AD480395

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

**Distribution authorized to U.S. Gov't. agencies and their contractors;
Administrative/Operational Use; 1961. Other requests shall be referred to U.S. Naval Postgraduate School, Monterey, CA 93943.**

AUTHORITY

USNPS ltr, 6 Oct 1971

INFORMATION RETRIEVAL

A. MARTIN WILDBERGER

INFORMATION RETRIEVAL

* * * * *

by

A. Martin Wildberger

INFORMATION RETRIEVAL

by

A. Martin Wildberger
//
Lieutenant, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

IN

MATHEMATICS

United States Naval Postgraduate School
Monterey, California

1961

INFORMATION RETRIEVAL

by

A. Martin Wildberger

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

IN

MATHEMATICS

from the

United States Naval Postgraduate School

ABSTRACT

This paper is divided into two distinct parts. The first is a summary of the general theory of information retrieval. A comprehensive mathematical model is described in terms of the theory of Boolean lattices, which serves to unify and make precise the basic problem of information retrieval. All possible basic methods of coding information for storage and retrieval are briefly described and contrasted. Another mathematical model for information retrieval based on linear graphs and stochastic processes is briefly described as an alternative to the lattice model. The appendices contain a survey of lattice theory, and an example of superimposed coding.

The second part of this paper is a detailed example of the application of information retrieval techniques utilizing the facilities of the USNPGS Computer Center to handle a problem involving the technical reports section of the school library.

The writer wishes to express his appreciation for the assistance and encouragement given him by Professors Elmo J. Stewart, Willard E. Bleick, and Edward Ward of the U. S. Naval Postgraduate School in this investigation.

TABLE OF CONTENTS

PART ONE: THEORY OF INFORMATION RETRIEVAL

PART TWO: A SEMI-AUTOMATIC BIBLIOGRAPHIC

INFORMATION RETRIEVAL SYSTEM

LIST OF ILLUSTRATIONS

PART ONE

THEORY OF INFORMATION RETRIEVAL

I

INTRODUCTION

1. Definition and Scope

Part One of this paper will deal chiefly with the theory of information retrieval in general. However the discussion will be restricted to the extent that those problems will be emphasized for whose solution large commercially available electronic computers are readily adaptable. We define information retrieval as an operation performed on a stored file of individual items containing coded or classified descriptions of their referrent. The operation consists of selecting those items which satisfy a given set of search criteria and then presenting the individual references to the searcher. In electronic computer systems, for instance, the file might consist of magnetic tape on which are stored coded individual items. The retrieval process then would consist of a sequence of operations under computer control designed to select automatically every item which meets all search criteria. As an adjunct to the search process, the reference portion of the retrieved item might be machine edited and printed in some convenient form. The overall system might further undertake to automatically process the file itself by deleting and/or correcting old material and adding new. Recent experiences have shown that the use of electronic computers can provide fast, accurate, convenient, and inexpensive retrieval. [1]

There is a very wide range of information retrieval problems, however, and all of them are not suitable for handling by electronic

computers. Three determining factors may be said to be: the size of the information file, the frequency with which questions are posed for retrieval, and the complexity of the criteria by means of which desired information is selected. It so happens that, just as the increasing magnitudes of these three factors point toward the utilization of a large digital computer, so also do they point out the need for an overall, unified theory from which the problem of information retrieval may be attacked. [2]

2. Criteria and Approach

Before proceeding to discuss possible approaches to such a theory, we would first like to consider what one should expect from a satisfactory information retrieval system. The reason who will use an automated information retrieval system by asking questions and receiving reports will be primarily interested in accuracy and speed. But he would also prefer a system which will make few demands on him as far as learning new techniques is concerned, and he would somehow like to reserve the privilege of browsing through the file if it were possible, since one function of a collection of documents is to stimulate new and unexpected approaches to his problem.

From the librarian's or documentalists' viewpoint the daily routine activities necessary to operate the system must be performed with the utmost of convenience. Therefore, the manner in which the searching and file maintenance entries are prepared must be as simple and direct as possible.

A customer desiring to use an information retrieval system

actuates it by presenting a "prescription" for the information that he wants. The retrieval system responds to this prescription by indicating to the customer a set of documents from the collection which presumably will furnish the information he desires. In other words, an information retrieval system translates or transforms the customer's prescription into a set of documents. [3]

From this operational point of view we shall begin, in the next chapter, the construction of a mathematical model for the information retrieval problem.

II

A LATTICE MODEL FOR DOCUMENTS[1]

1. Sets of Documents

In the following model, we shall consider a library to consist essentially of a collection of n documents (books, pamphlets, periodicals, etc.) which we shall call U. Every "batch" of these, selected by any means from among the whole collection will be called a set of documents. An example of such a set would be "all documents (in the library) written by J. G. Jones." Another example would be "all documents (in the library) bound in red vellum." We say that two such sets are identical if they contain precisely the same documents. Clearly any possible such set can be defined by enumerating its contents, i.e.; by submitting a list of documents by their "call numbers" [4] (or even by title if there are no duplications of titles in the whole collection.)

Now consider all possible such sets of documents taken from the library. The aggregate of all such sets is a new collection, a set of sets, which we shall call L. L has 2^n distinct members including the null set 0, containing no documents, the set of all documents U; n sets each containing one document; and in general $\binom{n}{m}$ distinct sets consisting of precisely m documents (for $m < n$). Thus for each choice of documents that could be taken from the library, there exists a member of L. A moment's reflection will show that

[1] In this chapter and the next references will be made to definitions and theorems in Appendix A.

most of the members in L represent heterogeneous sets of documents with no unifying similarities in their subject content; such sets are not a useful output to any input retrieval request. This fact constitutes a central problem in information retrieval.

2. Requests for Information

A request for information from a library will be viewed in this model as a "prescription" consisting of logical constraints which describe a desired set of documents. However the set need not actually exist in any library. We shall have occasion to refer to the collection of all possible such requests, denoted by R, and we shall refer to an individual request as a member of R.

The purpose of any retrieval system is to receive as an input a request or query which can be represented by a member in R, abd to convert this input to an output which consists of a citation to, or a set of copies from, some set of documents which is in turn represented by a member in L. Any retrieval system defines a transformation or mapping T, which takes each member r in R onto some member l in L. In order to characterize such mappings, and to select the most useful one in a particular application, it is first necessary to study the structure of the aggregates R and L, both as algebraic systems and as topological spaces. We have seen how the individual documents in a library may be thought of as distinct elements, for they are uniquely distinguished at least by their respective call numbers. But they are not ordered in any sense except arbitrarily by, say, location. Any set of documents from a

library, i.e.; any set in L; is definable, in general at least by enumeration. Two such sets are distinct if one contains at least one document (element) not contained in the other. They may be partially ordered (DEF.II) by inclusion, but they are not, in general, linearly ordered (DEF.III) since some at least are disjoint. Any subset (DEF.I) of members of L has a greatest lower bound (DEF.VI), namely the intersection of its member sets, which is also a member of L. Every subset of members from L has a unique least upper bound (DEF.V), namely the union of its members, which is also a set in L. Therefore L is a finite lattice [5] (DEFS.VII). In fact L is a Boolean (DEF.XIII) algebra under the usual set of theoretic operations of union, a+b, intersection, ab, and completation a'.

3. Classification Systems

From the sets of L definable by enumeration, any system of classification serves simply to select certain distinguished subsets of L. The various methods of classification for cataloging documents have but one characteristic in common, i.e., they are exhaustive in the sense that each document lies in at least one of the distinguished subsets of L.

Now let us examine the structure of the subsets distinguished under various methods of classification of documents.

Source

We begin with the simplest case of classification by source, i.e., by publisher or by author. In the latter case, each subset is distinguished by the name of an author, and consists of all the books in the library which were written by the designated author. These distinguished subsets have no ordering except an arbitrary one such as alphabetic. They are all disjoint, provided we make the convention that a joint authorship subset does not include the subsets of its individual authors, but only contains those documents written by the group jointly. If we denote the set of subsets distinguished as to source by A, then A^*, its closure under union, intersection, and complementation, is indeed a Boolean lattice and a sublattice of L. (DEF.IX).

Date

Now consider classification by date of publication. This could be viewed as the same as that by source, that is; in the sense "all documents published on said date". However, it is more interesting and useful to think of this as a classification in terms of "since said date" or "before said date." Either system distinguishes subsets of L which form a (ascending or descending) chain (DEF. XV) in L, (D_s, since date; or D_b, before date.) which is a sublattice of L. Either D_s or D_b alone is strongly distributive (DEF. XV) but not Boolean. It requires the union of both, $D_s + D_b = D$, to contain all complements (DEF. XII) and to form a Boolean algebra.

Uniterms

The system of classification known as coordinate indexing selects distinguished subsets of L by the use of descriptors or uniterms which are words, word roots, or short phrases which describe, in part, a document's contents. In theory, any number of uniterms may be assigned to a particular document, and the list from which they are taken may be either permanently fixed or open-ended. Let G be the set of all subsets each of which is distinguished by a single descriptor. Then G is partially ordered by set inclusion. But G is not a lattice since the union of those documents relating to one term, g_1, with those documents relating to another, g_2, i.e; $g_1 + g_2$, does not always distinguish a set, g_3, already selected by some other single descriptor. However, if we take the closure (DEF. XIII) of G, G^*, then we obtain a Boolean sublattice of L which contains all sets distinguished by any logical combination of descriptors. One school of thought on coordinate indexing [6, 7] considers it highly desirable that no set of documents distinguished by a particular descriptor be identical to or wholly contained in another set distinguished by a different descriptor. If this condition holds, then all the irreducible elements (DEF. XVII) of G^* are precisely all the documents and each is representable by an irredudant intersection (DEF. XVIII) of members of G. In fact, every member of G is a point (DEF. XIX) in G^*, and conversely, every point in G^* is a member of G.

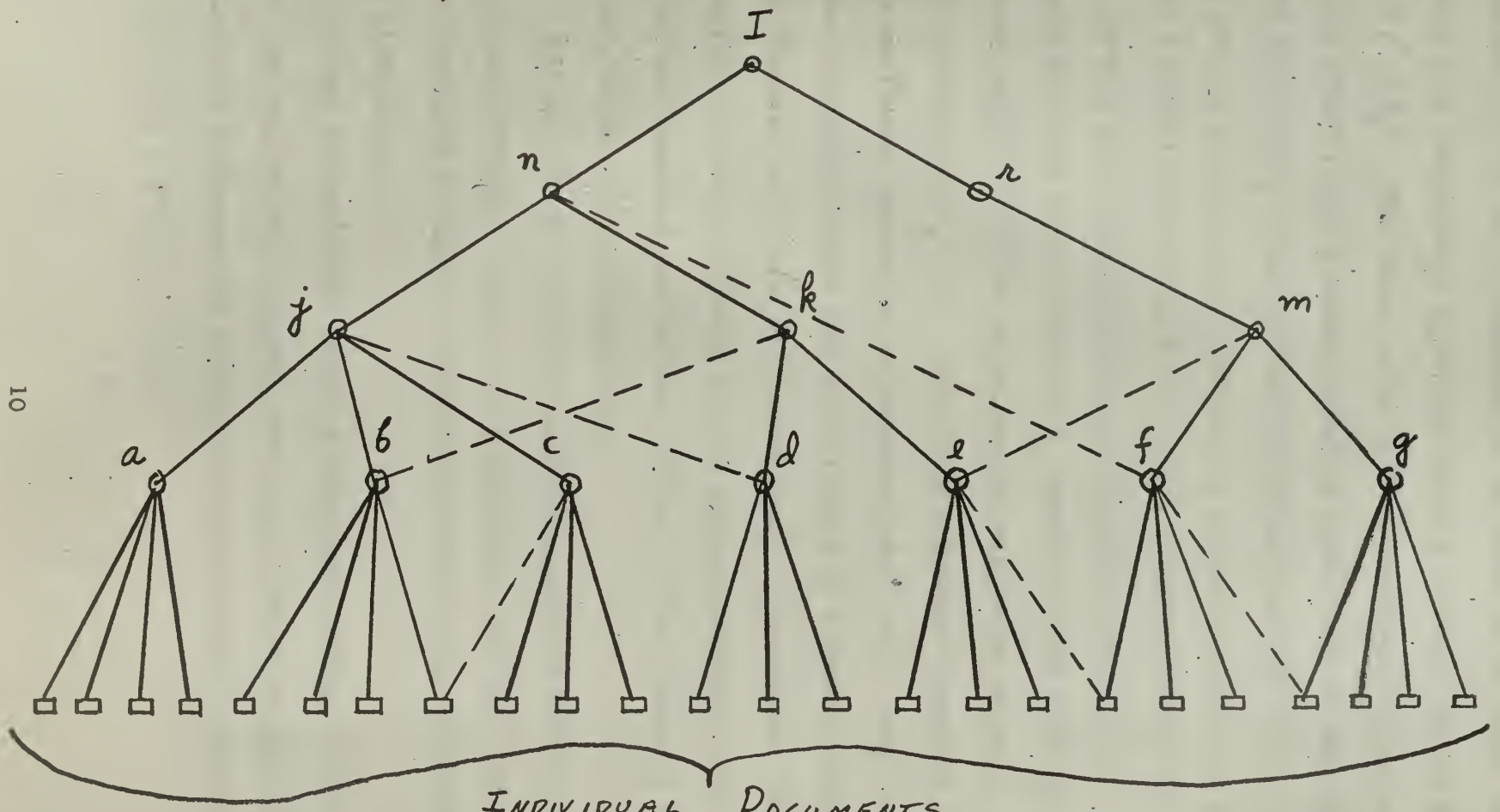
Subject Heading

The most commonly used and most complex method of classification employs a hierarchy of subject headings [4]. There are two basic kinds [3]. In a strictly hierarchial system no cross references are permitted; the linear graph of such a system is a tree such as that shown in figure 2.1 (ignoring the dotted lines.) The sets of L distinguished by a strictly hierarchical classification form by themselves a lattice denoted by H, which is however, not a sublattice of L. H is itself closed under intersection and union and is strongly distributive. But for h in H, $h' = I - h$ is not in general a member of H. Hence H is not Boolean. The structure of H is unsatisfactory for most retrieval purposes for another reason. Consider set a through r in H (refer to figure 2.1), note that $a \leq j$, $d \leq k$, but $jk=0$. Then the least upper bound of j and d in H is $j+d=n=j+k \geq e$. On the other hand their l.u.b. in L is $j+d=1$ some member of L not in H. Normally, if we request the logical disjunction of the sets distinguished by the two concepts d,j, that is; "j ord," we mean the set 1 rather than n for we do not wish to include e. This difficulty arises from H not being a sublattice of L. Thus it is necessary to consider the closure of H, namely H^* which is a (Boolean) sublattice of L.

The other type of hierarchical structure which permits cross references is called weakly hierarchical. Such a system is shown in figure 5.2 or in figure 2.1 if we include the dotted lines. Consider the subset H_w of sets of L distinguished by a weakly

FIGURE 2.1

LATTICE OF HIERARCHICAL CLASSIFICATION



hierarchical classification. Let H be the subset of sets of L distinguished by the same classification exclusive of its cross references. H is contained in H_w and the closure of either is the same, i.e.; $H^* = H_w^*$. What then, is the lattice-theoretic distinction between them? From a graphical point of view it may be seen that there may be in H_w, more than one path between an individual document and the vertex I. Any such path is a chain of sets, but since H_w is a modular (DEF.X) (in fact strongly distributive) lattice any two such chains have equivalent refinements (DEF.XIV and Theorem IV). One of the chains in H_w between a given individual document and I will always be the unique chain which exists in H, hence any chain in H_w (i.e., any path to a document through H_w) has a refinement equivalent to the unique chain in H. Only if we conduct a stepwise search does the existence of a chain of possibly lower dimension in H_w gain us any advantage. If we require the flexibility to handle searches for all logical functions of the various distinguished sets we must still go to the closure of the lattice, which is H^* in either case.

This conclusion is not very satisfying since we feel intuitively that cross references should be of considerable more use. The maze model discussed in Chapter 4 attempts to utilize cross references more effectively.

As far as the lattice model is concerned, however, it has been shown that any standard classification system which distinguishes certain sets from among all possible sets of documents, generates as its closure a Boolean lattice.

4. The Catalog Card

Now we will look at the catalog card or other clerical record which represents an individual document. We shall take the viewpoint that such a record is essentially a logical function which relates subsets of L distinguished by the same or different methods of classification. This is best explained by an example.

Figure 2.2

Typical Library Catalog Card

SEMI-CONDUCTORS
E8151 D3

DeFrance, Joseph J.

Electron tubes and semiconductors.
Englewood Cliffs N.J., Prentice-Hall,
1958.
288p. illus 9x6in.

ELECTRON TUBES

In figure 2.2 there is an example of a library catalog card. This card classifies the book it represents as to subject (two headings) author, publisher, date and place of publication, number of pages, size and illustrations. The implication is that this document is simultaneously a member of a particular set distinguished by each of the several methods of classification [4]. For instance, it is both on the subject of semi-conductors and authored by J. J. DeFrance. We have here the intersection of sets

distinguished by different methods of classification. For instance, the product ha where h is in H_w and a is in A. We are required to consider the set of all possible such products, and hence the cardinal product of H_w and A. In the practical case, it is unnecessary to consider any other logical operation among members of different classification systems. For instance, a catalog card would not normally specify that the volume in question was either published by Prentice-Hall, or else had 288 pages, or both. This is reflected by the model in that only the cardinal product (DEF.XX) of two lattices is again a lattice. (Theorem IX). Neither the cardinal sum nor any of the ordinal operations preserve all the structure we require.

If we consider the Boolean closure of the cardinal product of two or more classification systems, it is identical with the cardinal product of the Boolean closure of the several systems, e.g.: $(HA)^* = H^*A^*$. (Theorem IX).

We can now define the Boolean lattice B as the cardinal product of the several Boolean lattices generated (as described above) by the various classification systems used in the composition of the library catalog card. In B, every document is a point (DEF.XIX). However, a catalog card is, in general, only a redundant representation of the document it describes. For instance, in one example, it may happen that the author, DeFrance, has never had a book published by any other company than this one, i.e.: Prentice-Hall. Then the set distinguished by "all books published by Prentice-Hall" may be

eliminated from the representation of this document without permitting that representation (as the intersection of subsets) to include any other documents. In this regard, classification by pagination and place of publication, for instance, are almost always redundant.

5. Summary

To summarize thus far: each document in a library may be represented as a point in a Boolean lattice which is the direct product of the closures of the sets of subsets of documents distinguished by the various systems of classification employed. In the following section we turn our attention to the space of requests for information from the library and define a mapping (the retrieval function) between it and the space of documents.

III

A LATTICE MODEL FOR RETRIEVAL

1. The Idealized Request

We have previously referred to the users request for information as a prescription. Of course it is not always presented in that way. It may be vague and even self-contradictory. This gives rise to a basic and very difficult task which is also central to the automatic translation of languages, that is: by what rules may we so formalize all human communications as to make their meanings always unmistakable? [2] .

This problem is beyond the scope of this paper and it will be bypassed by the following assumption. We assume that any raw query can be converted or transformed by unspecified operations into an "ideal" request which obeys the rules of formal logic and has a prescribed and predetermined format. We shall take the approach that such an idealized request may be represented by a logical combination of constraints on possible classification systems. [8] We shall call these constraints admissible if they embody logical operations conforming to the inherent structure of the particular classification system to which they relate. Otherwise they will be called inadmissible. Several examples follow based upon classification systems already described in Section 2.

In the case of classification by source, the most elementary constraint would take the verbal format: "All documents from

source a" or symbolically simply "a". Then admissible operators within the classification by source are disjunction and complementation but not conjunction. (See tables in figures 3.1 or 3.2) Recalling the convention adopted for documents having multiple authors, we note that a document cannot normally be from two or more sources simultaneously.

In the classification system using descriptors (set G), all logical operations are admissible, but we shall adopt the convention, (see Chapter 2, Section 3) that no descriptor will be admissible if its presence always implies the presence of another (admissible) descriptor.

In the case of hierarchical classification even this restriction must be dropped and all possible logical operations give rise to admissible constraints.

For a summary of the admissible and inadmissible constraints for several classifications see figures 3.1 and 3.2.

For somewhat the same reasons discussed in the case of the document record or catalog card in Section 2 we shall restrict logical operations between the various classification systems to that of conjunction (both-and) only. This is not a significant restriction in practice bwcause a raw request involving disjunction (either-or) between constraints on different classification systems may be broken into two or more separate ideal requests.

Figure 3.1 TABLE OF ADMISSIBLE CONSTRAINTS FOR IDEALIZED REQUESTS

CLASSIFICATION SYSTEM	ADMISSIBLE CONSTRAINTS	VERBAL COUNTERPART OF ADMISSIBLE CONSTRAINT
A (by source)	a + b	all documents from either source a or source b
	a'	all documents except those from source a
D_s (by date)	d	all documents published since date d
	d'	all documents published before date d
	de' where $e \geq d$	all documents published since date d but before date e
G (by descriptors)	g (unless $g \Rightarrow k$)	all documents associated with concept g
	g'	all documents not associated with concept g
	gk(g,k in G)	all documents associated with both concept g and concept K
	g + k	all documents associated with either concept g or concept k
H or H_w (by hierarchical subject heading)	all logical operations are admissible.	

Figure 3.2 TABLE OF INADMISSIBLE CONSTRAINTS FOR IDEALIZED REQUESTS

CLASSIFICATION SYSTEM	INADMISSIBLE CONSTRAINTS	REASON FOR INADMISSIBILITY	VERBAL COUNTERPART OF INADMISSIBLE CONSTRAINTS
A (by source)	ab	$ab = 0$	all documents simultaneously from both source a and source b
D_s (by date)	de	$de = \begin{cases} d \text{ if } d > e \\ d \text{ if } d = e \\ e \text{ if } d < e \end{cases}$	all documents published both since date d and since date e
	d + e	$d + e = \begin{cases} e \text{ if } d > e \\ d \text{ if } d = e \\ d \text{ if } d < e \end{cases}$	all documents published either since date d or else since date e
	de' where $e < d$	$de' = 0$ if $e < d$	all documents published since date d but before date e as well, where e is earlier than d
G (by descriptors)	g if $g \rightarrow k$ where k is in G	$gk' = 0$	all documents associated with concept g where g implies k and k is also a member of G
H or H_w (by hierarchical subject heading)	no logical operation is inadmissible.		

2. The Space of Requests

Let every possible ideal request be fulfilled conceptually by at most a denumerable number of possible documents. Note that an impossible document in this sense would only be one which incorporated a logical constradiction, but we have eliminated the description of such "impossible" documents by permitting only admissible constraints in our ideal requests.

First, we consider the set of all possible documents, $\underline{U}$. Note that $\underline{U}$ is infinite but denumerable. This may be viewed as a hypothetical library, which would yield an infinite number of appropriate documents in response to every conceivable request for information which we might make. Again, as in the case of the actual library, we consider the collection, $\underline{L}$, of all possible sets of members of $\underline{U}$. Conceptually, at least, the members of $\underline{U}$ may be classified as to author, date, subject, etc., resulting in the denumerably infinite partially ordered sets and lattices $\underline{A}\ \underline{D}\ \underline{G}\ \underline{H}$ ... and $\underline{A}^*\ \underline{D}^*\ \underline{G}^*\ \underline{H}^*$ ... As in the finite case, each of these is generated by the members of $\underline{L}$ distinguished by a particular system of classification. In this space of possible documents, the idealized request is somewhat analogous to the catalog card. These requests may be considered as generating a lattice similar in most ways to B, described in Chapter II. However, there are significant differences.

Unlike the catalog cards in the space of actual documents, the requests are not necessarily points in $\underline{R}$. Indeed, one request may contain many others, and the same possible document may fulfill all

the requests in a chain. The space $\underline{H}$ is a lattice, the cardinal product of several lattices, not all of which are Boolean (because of the restriction to admissible constraints.) $\underline{R}$ has a denumerable infinity of elements, the possible documents. It is required to map $\underline{R}$ onto the space B of sets of actual documents, which is a cardinal product of Boolen lattices with a finite number of elements.

3. The Retrieval Homomorphism

The mapping is a homomorphism (DEF.XXI) by $T: \underline{R} \rightarrow B$. In what follows, components (DEF.XX) of $\underline{R}$ will be denoted by underlining. If we exclude the inadmissible constraints as logical operations we must define T differently for the various components in the direct products $\underline{A}\ \underline{D}\ \underline{G}\ \underline{H}\ \ldots\ \underline{R}$ and ADGH....= B. For instance, $T: \underline{A} \rightarrow A$ is a join-homomorphism only and $T: \underline{D} \rightarrow D$ is a homomorphism of the two spaces as semi-groups only.

On the other hand, $T: \underline{H} \rightarrow H$ is a lattice-homomorphism between their respective closures. In this case we may look at $\underline{H}$ as including H, since all actual subject headings are included among all possible ones. Then a particularly simple characterization of T is as the equivalence relation on $\underline{H}^*$ which generates H^* as its convex sublattice of ideals (DEF.XXII and Theorem X). This characterization of T can be extended to all components of the direct products if the inadmissible constraints are re-admitted using as their operational definitions the expressions appearing in column three of figure 3.2. Now it is possible to describe B as the direct product of convex

sublattices of $\underline{R}^*$ generated by the homomorphism T taken as an equivalence relation on $\underline{R}^*$ (Theorem XI). Each request in $\underline{R}$ defines a direct product of ideals in B, one from each component lattice.

In terms of this structure, it is possible to describe most of the verbal statements concerning the properties of classification and retrieval systems which are necessary or in some sense desirable. For example, those requests in R which cannot be filled are the analog of zero in B.[3] Also redundancy among several classification systems may be expressed by stating that the center (DEF.XXIII) of B is not empty, for in an irredundant system of classifications the maximal distributive sublattice of B are in fact the several components of the direct product. (Theorem XII).

4. Summary

In summary, we have seen that the retrieval function may be defined as a lattice homomorphism of the set of all possible ideal requests onto the set of subsets of documents, $T:R^* \longrightarrow B$. Moreover, each request defines a direct product of ideals in B, one from each classification system under which the documents are catalogued.

IV

CODING FOR INFORMATION RETRIEVAL

1. Definitions

Every material aid to the storage and retrieval of information, from the catalog card to the digital computer requires that the information it handles be put in some particular format or language. [9] And as the assisting apparatus grows more complex, its natural language seems to depart the further from that of the humans it serves. Therefore, a part of the problem of information processing is that of translation from one language to another in the most effective and economical manner. This is what we will call "coding". 10 We shall first describe three basic types of codes which have been used in information retrieval systems. Discussion of their advantages and limitations will be put, as much as possible, on a mathematical basis. [11] Secondly, we shall attempt to gain some insights into the problem of coding from information theoretic considerations.

For simplicity, and because the vast majority of data processing devices use binary arithmetic, we will discuss coding in terms of binary digits or "bits".

The coded information may be thought of as being represented by charged or non-charged spots on magnetic drums or tape, holes in cards, paper tape, or for that matter, notches in a post.

2. Direct Coding

A direct code, by definition, uses a single bit (or notch) to represent a single idea. In this system, if we have H different ideas

to code, we will require H bits in each record. Or, to state the matter in a different way, if our record is planned to contain H bits (either bit either 1 or 0), direct coding will force us to analyze our subject matter in terms of not more than H concepts. Direct coding of itself cannot cause the searching operation to produce extra records. Records selected by testing for 1 in a given bit position have been coded for a given concept - no more and no less. Within the limits imposed by the restricted number of concepts in terms of which subject matter can be analyzed when using direct coding, it affords the simplest and most convenient method for conducting direct search operations. A single run through by a machine of limited memory is all that is needed.

3. Selector Coding

Direct coding, as we have seen, attaches meaning to a single bit position. It is also possible to attribute significance to a combination of bit positions. When this is done in such a way as to minimize the amount of searching in order to isolate records characterized by some one combination of bits in a particular field, the resulting scheme is termed a "selector code".

If we let C denote the number of combinations of H things taken Y at a time, then

$$C = \frac{H!}{Y!\ (H-Y)!} \tag{4.1}$$

For example, if we attach meaning to a combination of two bits in a field of size five, then we can indicate any one of ten concepts

in that field, as we find by substituting in equation (4.1)

$$C = \frac{5!}{2!(5-2)!} = \frac{1.2.3.4.5}{(1.2)(1.2.3)} = 10$$

If the value of Y in equation (4.1) were unity, then the code would revert to the direct type. From a mathematical point of view a direct code is the limiting case of selective coding.

The maximum number of combinations for fixed field size, H, is obtained when

$Y = \left[\frac{1}{2}H\right]$, the largest integer equal to or less than $\frac{1}{2}H$.

4.2) $$C_{max} = \frac{H!}{\left[\frac{1}{2}H\right]! \left[\frac{1}{2}H-Y+1\right]!}$$

Thus, with a fixed number of spots in a field of size eight the maximum number of combinations is 70.

$$C = \frac{8!}{4!(8-4)!} = 70.$$

In order to evaluate the usefulness of selector codes it must be kept in mind that they permit only one concept to be coded in each field. Hence, selector codes are principally useful for coding a single member of a group of mutually exclusive concepts (disjoint classes) of which the date of publication and the name of the publisher are good examples. Selector codes have found their greatest use in edge punched cards where the sorting is accomplished by running needles through the holes of the field in question. Since fewer holes are required (than in direct coding) for the same number of coding possibilities, selector coded cards can in general be arranged in sequence with less effort.

4. Sequence Coding

Minimum effort in sorting records into a predetermined sequence is the chief advantage of sequence codes. Sequence codes, like selector codes, are based on the principle of attributing meaning to a combination of bits in a fixed field. But in this case the number of bits to be used is not fixed. Sequence codes are based on taking all possible combinations by letting Y vary from zero to H. They make available for subject-matter analysis a number of concepts equal to the sum (C_s) of a series of selector codes (C).

$$(4.3) \qquad C_s = \sum_{Y=0}^{Y=H} \frac{H!}{Y!(H-y)!} = 2^H$$

A sequence code offers many more coding combinations than a selector code. Thus, in a size eight field, we have $2^8 = 256$ possibilities vice 70 for selector coding. However, it is still true that only one combination of the sequence code in question can be punched in any individual field.

It may be shown that searching for any one combination punched in a sequence-coded field of size H will require that all H positions be examined. On the other hand, in a selector coded field, an item is identified precisely once the Y one's have been located, and only H-1 positions need ever be examined.

5. Coding Efficiency

Up to now we have considered only one coding field. In practice of course, combinations of fields are used, and an individual record

may be divided into a number of fields. However, with the types of coding discussed so far it is not possible to make more than one entry in a given field. Hence, we are limited to information which may be analyzed in terms of mutually exclusive concepts. We will now direct our attention to other coding methods not subject to the above mentioned restrictions.

First it is necessary to introduce the idea of efficiency of codifications. [127]

To determine how many symbol positions are required in an unambiguous codification using K different symbols, we consider each symbol as a number in the base K and use enough positions to count the number of items in the base K. That is, with N items or concepts and K symbols, the number of symbol positions in a codification must be no smaller than the smallest integer n such that

(4.4) $K^n \geq N$ (or $n \geq \log_K N$)

if the codification relating to an item is to be unique.

For the binary case which we have discussed so far, K = 2, and for 600 concepts, say, we have:

(4.5) $n \geq \log_2 600 = \frac{2.7782}{0.3} = 9.261$ or $n = 10$.

If more than n positions are actually used, the code is said to be redundant; if fewer than n, irredundant; and if exactly n, efficient. It is possible to utilize a purposely redundant code in order to detect and automatically correct errors in the recorded information introduced during storage or transmission. [13] However, the self-correcting

scheme that makes use of redundant positions must also be able to correct errors that may occur in the extra redundant positions as well. This self-correcting feature requires considerable additional operations to be employed by the data processing system. Hence, its use may be uneconomical from the point of view of operating time and equipment as well as the additional storage space required. The amount of redundancy necessary depends on the depth of error we desire to correct. It may be obtained by combining equations (4.3) and (4.4) to obtain:

$$(4.6) \qquad R = \log_K \left[\sum_{Y=0}^{Y=M} \frac{H'.}{Y'.(H-Y)'.} \right]$$

where M is the level of error correctable in a field of size H by the use of R redundant positions. Then the effective number of positions left for actual use in coding is H-R.

6. Superimposed Coding

On the other hand, if we have available a field of H bit positions, then there is, as noted above, room for $H = 2^H$ concepts to be coded efficiently. Suppose, however, we know that fewer than N of these actually appear in a particular situation. Furthermore, suppose that we are dealing with items each of which is associated with up to X of these concepts, then, at least $X \log_2 N$ bit positions must be used to identify an item, or X fields of the size H available. But we are here considering situations in which they do not all actually appear. The question arises: How

can we use fewer than X $\log_2$ N bits for coding, say H, and still distinguish among all N concepts?

By the superimposed code corresponding to X concepts, we mean the result of forming the logical sum of the individual sequence or selector codes for these X concepts.

This procedure permits the use of one field of size H in place of X such fields to code one item. However, we must evidently pay a price for this convenience and saving in item coding positions. For if items are to be selected from the file on the basis of fewer than X concepts, then it is possible for an item to be selected because the logical sum of the codes for the desired characteristics has units in positions which are among those in the superimposed code associated with a different combination of concepts. Appendix B contains an illustrative example of superimposed coding and also several formulas for the probability, p, that an item not associated with a particular characteristic will be selected during the search for that characteristic. The decision to use superimposed coding depends on the relative importance in a particular application of saving space versus the amount of false selections that can be tolerated. Note, however, that no item having the desired characteristics will be missed in a search merely because this technique was used in coding it. [14]

V

A MAZE MODEL FOR INFORMATION RETRIEVAL

1. Hierarchical Classification as a Maze

Another possible model for information retrieval systems is that of an abstract maze. [15] It is an analogy which may appear particularly apt to most library users. This interpretation is especially helpful in shedding light on the particularly difficult case of hierarchical classification with cross references. A portion of a subject catalog system linked by cross references is shown in outline form in Figure 5.1. The same area of the catalog is characterized graphically in Figure 5.2. The subject catalog achieves its maze attributes as a result of the sets of cross references linking the subject headings. Although it was developed originally as a search aid, the difficulty of keeping in mind much more than point to point search properties is a limitation to its usefulness.

2. Search Strategy

Given an initial subject heading related to the search requirements, the searcher would most certainly be helped in forming a strategy of search if he were able to study the character of the subject heading maze in a region around the point of entry.

Figure 5.1 Library of Congress Classification Scheme

(area around Mathematical Statistics)

025.4 U5 Q 1950

Q	SCIENCE
QA	MATHEMATICS
152 thru 297	ALGEBRA

273 Probabilities: Correlation. Discussion of observations.

Cf GV 1302, Games of Chance

HA 29-31, Theory of Statistics

HG 8781-8782, Actuarial Science

QH 405, Statistical methods (General Biology)

275 Theory of Errors. Least Squares

276 Statistics, Mathematical

276.5 Sampling (Statistics)

277 Graphic Methods for discussion of observations.

281 Interpolation

295 Series. Infinite products and other finite processes

297 Numerical calculations

QH NATURAL HISTORY

301-705 GENERAL BIOLOGY

401 Variations

405 Statistical Methods

Special results

406 Plants

407 Animals

411 Experimented Study

H SOCIAL SCIENCES

HA STATISTICS

29 Theory, Method

Cf QA 276 Mathematical Statistics

31 Graphic and mechanical methods

33 Other special

HG FINANCE

8781-8782 Actuarial Science. Theory of Probabilities.

Cf HG 8051-8059, Insurance

HJ 9711-9721 "Political Arithmetic"

QA 273 Probabilities

8781 General Works Treatises

8782 General special and minor eq. Survivorship.

GV RECREATION

1301 Chance and banking games. Gambling

Cf HV 6708-6722 Criminology

1302 Probabilities betting systems, etc.

1303 Dice and Dice Games

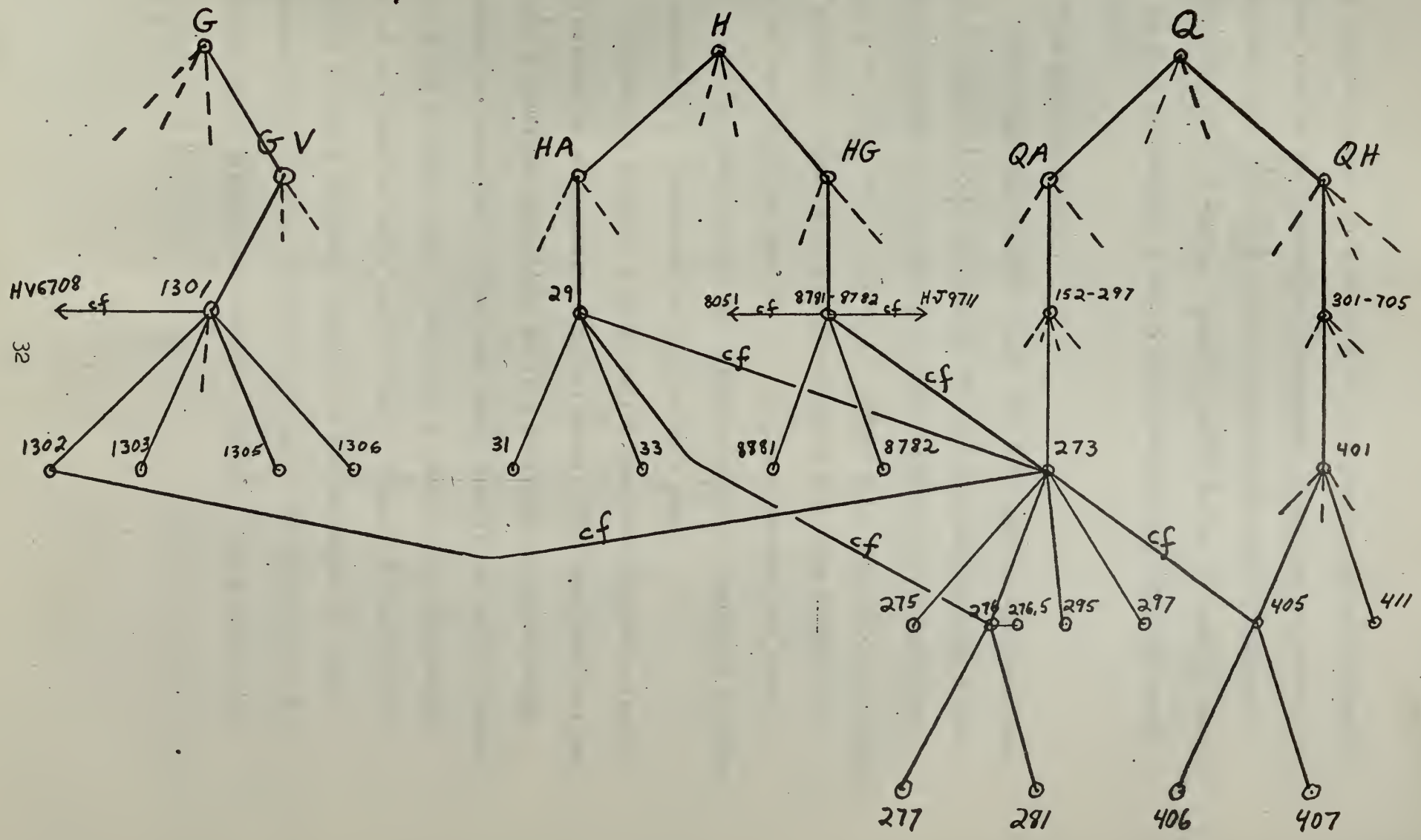
1305 Faro

1306 Keno

and others.

FIGURE 5.2

GRAPH OF SUBJECT CLASSIFICATION OUTLINE



In a machine search, this would appear to require pattern-recognizing devices such as the perceptron, or special techniques of topological classification, [16, 17] which are currently under development. However, there is a simple algorithm usable by man or machine which can be used to reorient a maze with respect to any arbitrary point of entry.

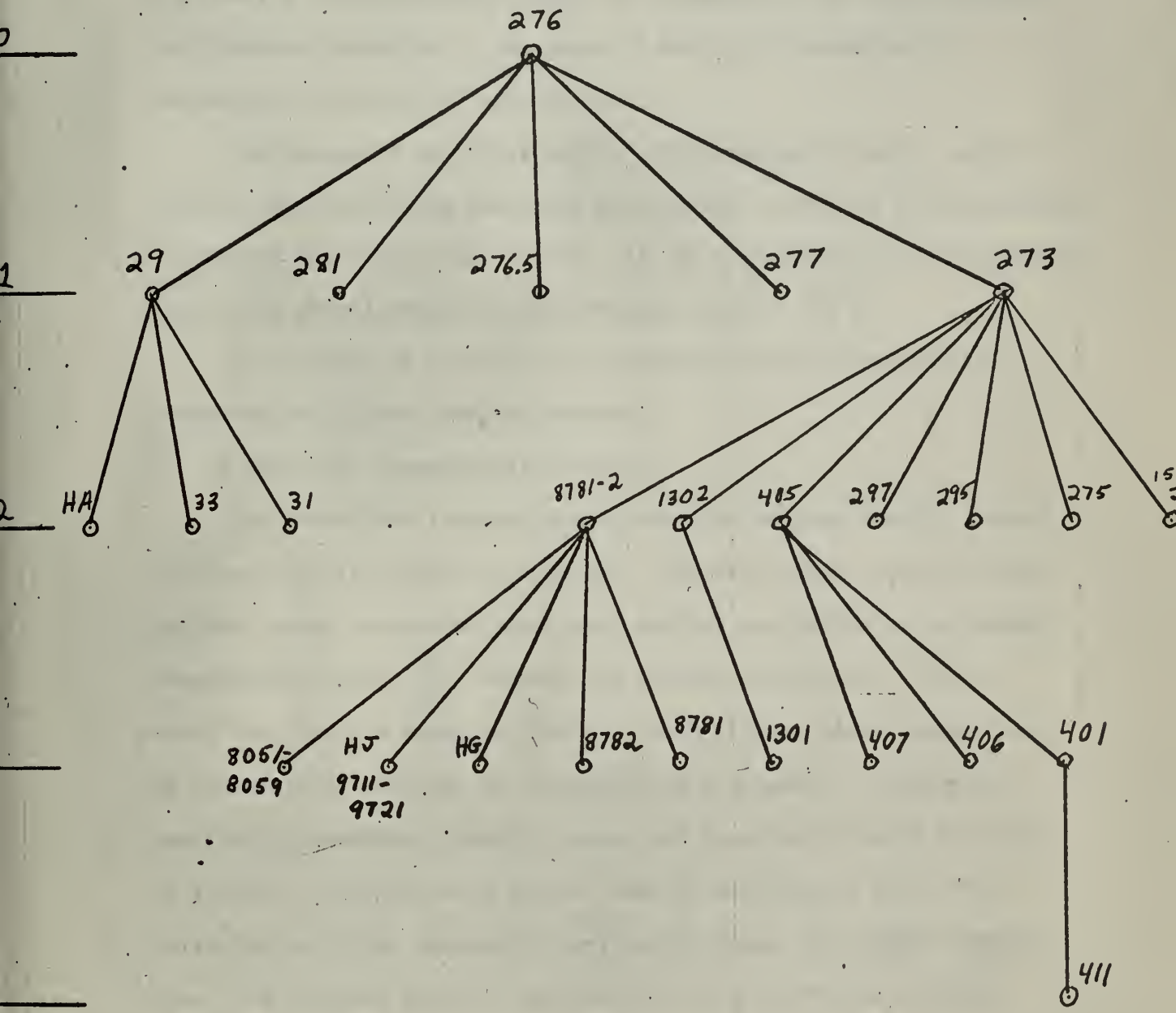
3. Maze Reorientation

This algorithm is essentially the first part of one developed by E. F. Moore [18] for finding the shortest path through a maze. This reorientation proceeds as follows: Select the origin point and tag it with a zero. Select all points which have not been tagged, but which are adjacent to the points which have been tagged and provide them with the tag $i'(i = 1,2,\ldots,k)$. Repeat this process until the k'th level points are tagged, where k is the depth to which it is desired to penetrate the maze on this pass.

Figure 5.3 displays a graph of the same catalog maze oriented with respect to a particular point of entry. The most practical use of such a procedure appears to be in a retrieval system allowing man-machine "cross-talk." The user could specify the point of entry or it could be made random. The machine would orient the catalog maze to his point of entry out to (say) k steps. It would then organize a search strategy according to the incidence of terms presented by the user to describe his goal. The search might be continued until a specified number of documents had dropped out.

FIGURE 5.3

GRAPH OF SUBJECT CLASSIFICATION REORIENTED TO POINT-OF-ENTRY



(These may be thought of as "dead end" lines in the graph.) The machine could repeat the reorienting algorithm as often as necessary, beginning anew at the one of the kth intersections characterized by the greatest number of the descriptive terms given originally. At any time, a reoriented graph might be produced for the users perusal and decision concerning a new point of entry, or broadening or narrowing of the search "prescription."

The suggested retrieval system, while requiring rather sophisticated data processing machinery would permit a sort of browsing which is greatly desired by most users. [1] It's employment in a particular case would depend mainly on cost considerations. [19]

In the case of a strictly hierarchical classification this technique has obvious simplifications.

4. An Empirical Classification System

The coordinate indexing system may also be described by a maze, but there is little advantage gained. However, there might be superimposed on the coordinate index some sort of statistically developed association trails [2] between the various descriptors. These would indicate the relative frequency with which a given combination of terms appear together in the record of a document. In such an empirically generated catalog system, all connections would be cross references, obtained not a priori from the descriptive terms but a posteriori from the documents described by them. The highly complex maze thus obtained would be susceptible to the searching procedure outlined above.

PART TWO

A SEMI-AUTOMATIC BIBLIOGRAPHIC INFORMATION RETRIEVAL SYSTEM

I

PHILOSOPHY AND PRELIMINARY CONSIDERATIONS

1. General Criteria

The first part of this paper has discussed the theory of information retrieval and suggested several mathematical models in terms of which this theory may be organized and described. The second part will be concerned with an experimental information retrieval system designed and put into actual operation as a practical example (and possible critique) of these theories. However, the design of such a system, while founded on the mathematical models, is as much an exercise in systems engineering as in mathematics. With this in mind, the following criteria [25] are proposed for use in the comparison of existing practical systems and also as constraints to be met in the synthesis of new information processing system:

1. Size of the file to be covered
2. Rate of growth of the file and system
3. Range of inquiries to be serviced, or the purposes to be served
4. Range of subject matter to be covered
5. Kinds of concepts to be represented
6. Specificity and type of analysis
7. Personnel required to do the analysis
8. Cost of processing information and conducting searches
9. Reliability of results, or probability of retrieval
10. Form of system

2. Specific Constraints

In the particular case of this example, further very practical considerations of economy, time, and human engineering imposed additional constraints. Of the existing library contents, classification and cataloging system, and clerical procedures, only the latter might be changed at all, and that as little as possible. The system could only be mechanized using the presently available equipment, and that only on a time sharing basis with many other projects of higher priority. On the other hand, the fact that no new equipment was to be obtained specifically for the system, permitted the use of trial and error as a legitimate improvement technique in the synthesis. Moreover the cooperation of those whose daily employment would be most directly effected by the system was outstanding, in marked contrast to nearly all reports on the introduction of automation into industry. [26, 27]

The technical reports section of the U. S. Naval Postgraduate School Library contains about 150,000 items and is growing at a rate of about 5,000 per year. The items vary in size from thin pamphlets to folios. They are stored in file cabinets, or are shelved individually or in boxes. About 60,000 have a security classification of confidential or higher and must be stored in a locked or guarded area. The reports are published by government agencies or by private institutions under government contract. They are of a scientific, engineering, or technical-operational nature.

On receipt the reports are assigned an accession or shelf number which locates them in storage but conveys no other information. They are cataloged as to source, report number (if any) assigned by the source, title, author(s), and date of publication. A sample of the catalog card is shown in figure 1.1.

Figure 1.1

Sample Catalog Card

U-46.301
Institute of Flight Structures, Columbia University
TN 1
Vibrations and stability of plates under initial stress, by G. Herrmann and A. Armenakas. February 1959

The only cross filing is by source. At the same time, the report is classified by being assigned any number of descriptors or uniterms each describing some phase of its contents, and its shelf number is entered on the card corresponding to each of these descriptors in a coordinate index file. The list of descriptors is open-ended in that new ones may be invented as the cataloger feels necessary. The descriptors may be words or word roots embodying general concepts; they may be technical terms; or even names of projects and weapons systems.

The reports are of an essentially transitory usefulness, which, however, may be measured in terms of months or years. Nevertheless, regular weeding of material obsolete in the judgment of the catalogers, is considered a virtually impossible task.

The technical reports library is used primarily bu graduate students in engineering who are writing research papers or perform-

ing laboratory projects in connection with establishing courses. A typical informal inquiry by one of them might be: "What do you have on plasma propulsion systems?" The library is also used by staff and faculty members who are more likely to have in mind a specific source or author; furthermore their advice to students as to where to obtain information on a particular subject usually takes this same form. In short, the inquiries tend to be either extremely general or extremely specific.

The equipment available with which to realize an automatic system consisted of:

CDC 1604	high speed digital computer	[28]
CDC 1607	magnetic tape input-output equipment	[29]
IBM	card punch	[30]
IBM	card to paper tape translator	[30]
IBM717 & 757	magnetic tape controlled printer	[30]
IBM402	accounting machine	[31]
Friden	"Flexowriters"	[31]
Remington-Rand	Synchro-tape typewriters	[32]

One of each of the last two items was available in the library itself. The rest were available on a time shaping basis in the USNPS Computer Center.

II

DESIGN OF SABIRS

1. Logical Design

The logical design of this Semi-Automatic Bibliographic Information Retrieval System (hereafter known as SABIRS) is based, in general, on the lattice model described in Part I. Each document in the library is represented by a record which is the logical product or intersection of the distinguished sets to which the document belongs under three different methods of classification: (a) by source, (b) by date of publication, and (c) by uniterms.

a is the set of all documents in the library originated by that particular agency or company.

b is the set of all documents in the library published during that particular month.

c is itself the intersection of n sets (n = 1 through 12) each one of which consists of all the documents in the library having to do with a particular uniterm.

In addition, each record carries as stag the shelf number of the document it represents. As an example, the document whose catalog card is shown in figure 1.1 would have the following record:

Source: Columbia University

Date: February, 1959

Uniterms: Elasticity, Stress, Plates, Motion, Stability.

Self Number: U-46,301

The raw requests for information by clients are idealized by being put into a standard form, an example of which is shown in figure 2.1.

Figure 2.1

Sample Request Form

NAME:	W. T. Door	
TELEPHONE:	Ex. 988	
ROUTING OR BOX #	1439	
SOURCE(S):	N.A.S.A.	00102064
	Lockheed Aircraft Co.	00100732
	Applied Physics Lab.	00100037
DATE OF PUBLICATION:	From March, 1960	DATE6003
	To present	THRU9999
UNITERMS:	Satellite	00006437
	Navigation	00003207
	Doppler	00000462

INSTRUCTIONS: If "any" source and/or date and/or subject is desired, leave corresponding code blank. No more than 15 total code-words may be entered. No more than 12 of them may be uniterms.

The only additional constraint on the formal request besides those described in Part I, Section 3, is that no disjunction of uniterms is permitted. Every such disjunction attempted requires the initiation of an additional request.

Each record on file has the following Boolean algebraic form: $s_j d_k u_l$ or, broken down to individual uniterms,

(21) $s_j d_k \prod_{i=1}^{n} t_i$, $n \leqslant 12$, where s denotes source, d denotes date, and t denotes uniterm.

Each admissible ideal request has the following Boolean algebraic form:

(2.2) $$\left(\sum_{j=1}^{m} s_j \right) \cdot \left(\sum_{k=j}^{k=q} d_k \right) \cdot \prod_{i=1}^{n} t_i, \quad n \leqslant 12, \; p \leqslant q$$

The retrieval system selects a set of documents from among those in the library. This set has a finite (possibly zero) number of members. That set represented by the request is denumerably inifinite. The correspondence is the homomorphism described in Part I, Section 3.

2. Functional Design

The functional design of SABIRS was guided primarily by the general philosophy and particular constraints described in Section 1. The three types of classification chosen were selected on the basis of past experience by the staff of the Technical Reports Library. It was considered highly desirable for a human operator

to be able to distinguish at a glance a source name from a uniterm in their coded form. Because of the rapid increase of publishing agencies, weapons system designators, etc., it was believed necessary to allow for at least 100,000 possible sources and uniterms. [5,33] It was convenient to utilize IBM binary coded decimal characters and to limit the length of a coded record or request to 120 such characters to allow direct use of the IBM 717 Line Printer. Furthermore, it was extremely convenient, because of the 48 bit word length of the CDC 1604 Computer, to make each code eight characters. Because of the high speed and large memory of the 1604, more economical coding was not considered necessary. Because of the semi-automatic nature of the system, with its frequent use of human intervention, easily decodable forms were considered highly desireable.

The present coded record of 15 eight-character words could easily be compressed if necessary; however it permits relatively systematic expansion of the capabilities of the system when the need arises. At present, over 75,000 records can be stored on one reel of magnetic tape. It is believed that future developments will be directed toward the storage of more information per record rather than toward physically shorter records.

Actual operating time on the CDC 1604 computer is held to a minimum by off-line preparation of input data on the Remington-Rand Synchro-Tape Machine and by off-line print out of the results on the IBM Line-Printer. It is estimated that actual computer

time for a daily run with a library of up to 150,000 records will be under five minutes. This includes the updating of the file of records which is accomplished at the same time as the search for information.

The master program for the 1604 phase of the system operation is a program generator type of data processing compiler rather than a formula interpretor. [34] This compiler is, of course, highly specialized but completely self-contained. It is believed that more future requirements will be able to be filled by additions to the executive program utilizing subroutines already available.

Considerable care has been taken to make the operation of the system straightforward and simple. The operating personnel are library staff members for whom some additional training is, of course, necessary; this consists mainly of enough practice, under instruction, to become facile in the operation of the machines. A wide leeway in input format is permitted before an error occurs in the output. There are a number of signals to indicate possible errors and, in addition, the input as read by the computer is printed as an output along with the results for cross checking. The following chapter contains a detailed description of the system operation, error signals, and particular capabilities and limitations of the system.

III

OPERATION OF SABIRS

1. Standard Operating Procedure

The routine operation of the Semi-Automatic Bibliographic Retrieval System is outlined in the flow charts of figure 3.1 (A through C). These should be followed through before reading the additional details listed below.

Starting the Main Computer Program

The search generator or compiler program for the Control Data Corporation 1604 Computer is stored on a reel of magnetic tape in assembly routine "AR" format. It must be entered into the computer memory from the tape unit designated four utilizing current operating procedures with the Computer Center's standard library of subroutines. With the compiler in memory and the most recent file of records on tape unit two, the reel of punched paper tape should be placed in the Ferranti reader and the latter set to "character" mode. After raising the start switch, the program will normally continue to completion without further intervention by the operator. Error signals which may appear on the consol typewriter are discussed in the next section. Upon completion, the computer stops at address 06037.

Interpreting The Output

The output from the computer appears on magnetic tape as shown in figure 3.1B. It may be printed as desired on the IBM Line-Printer. Examples of typical outputs and the inputs which generated them are contained in Appendix C The bibliography appears as a list of shelf

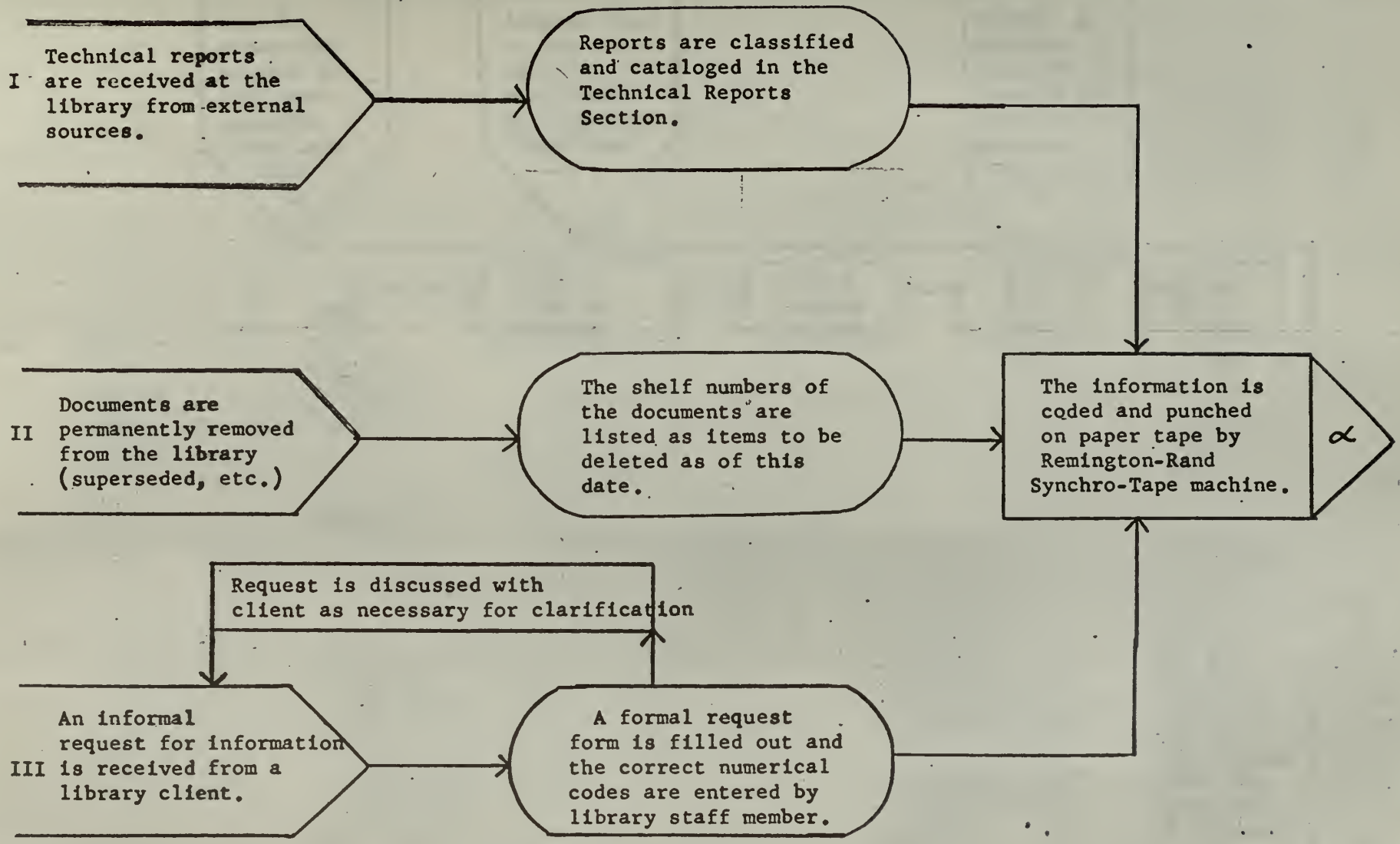


Figure 3.1A Operational Flow Chart for SABIRS

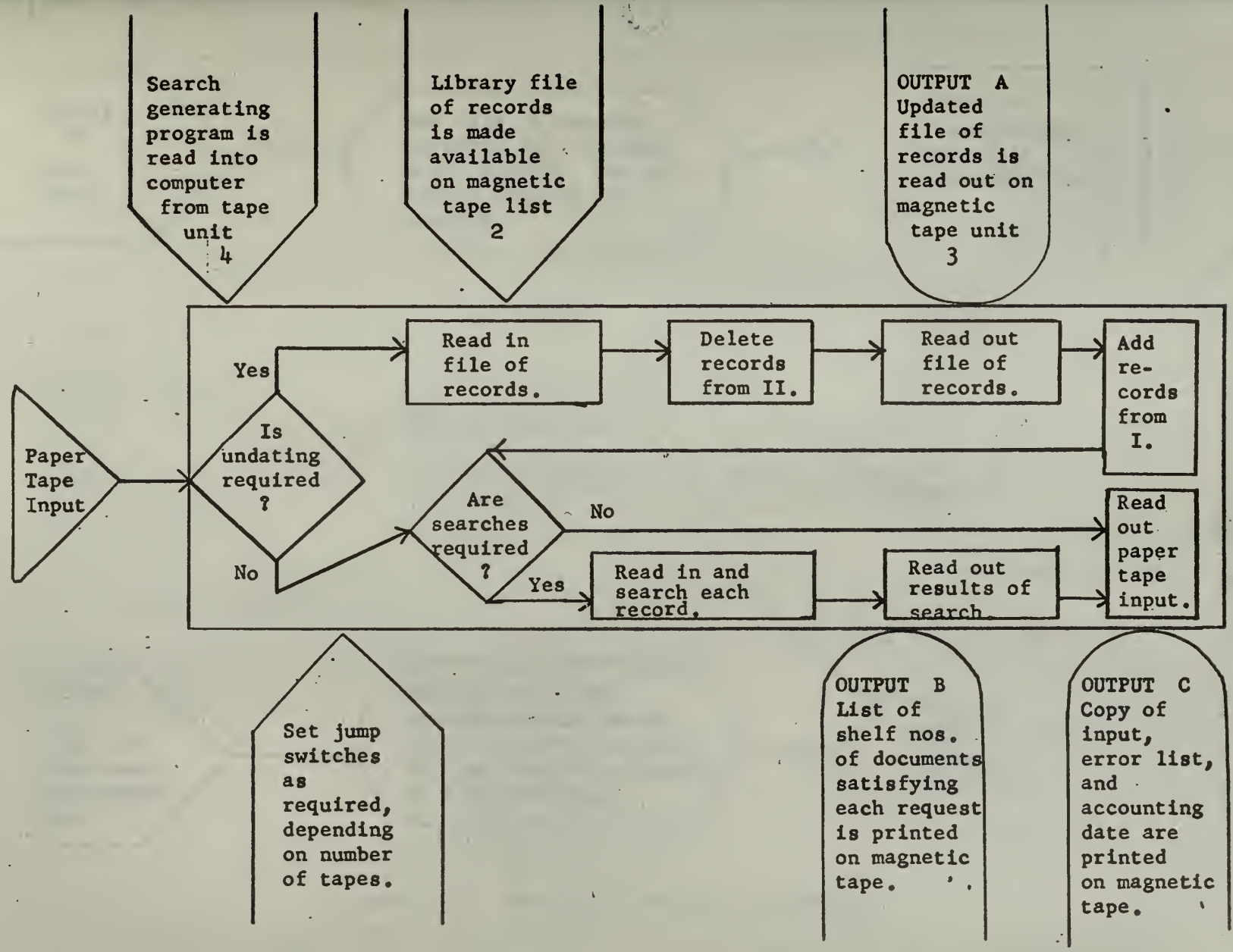


Figure 3.1B Operational Flow Chart for SABIRS

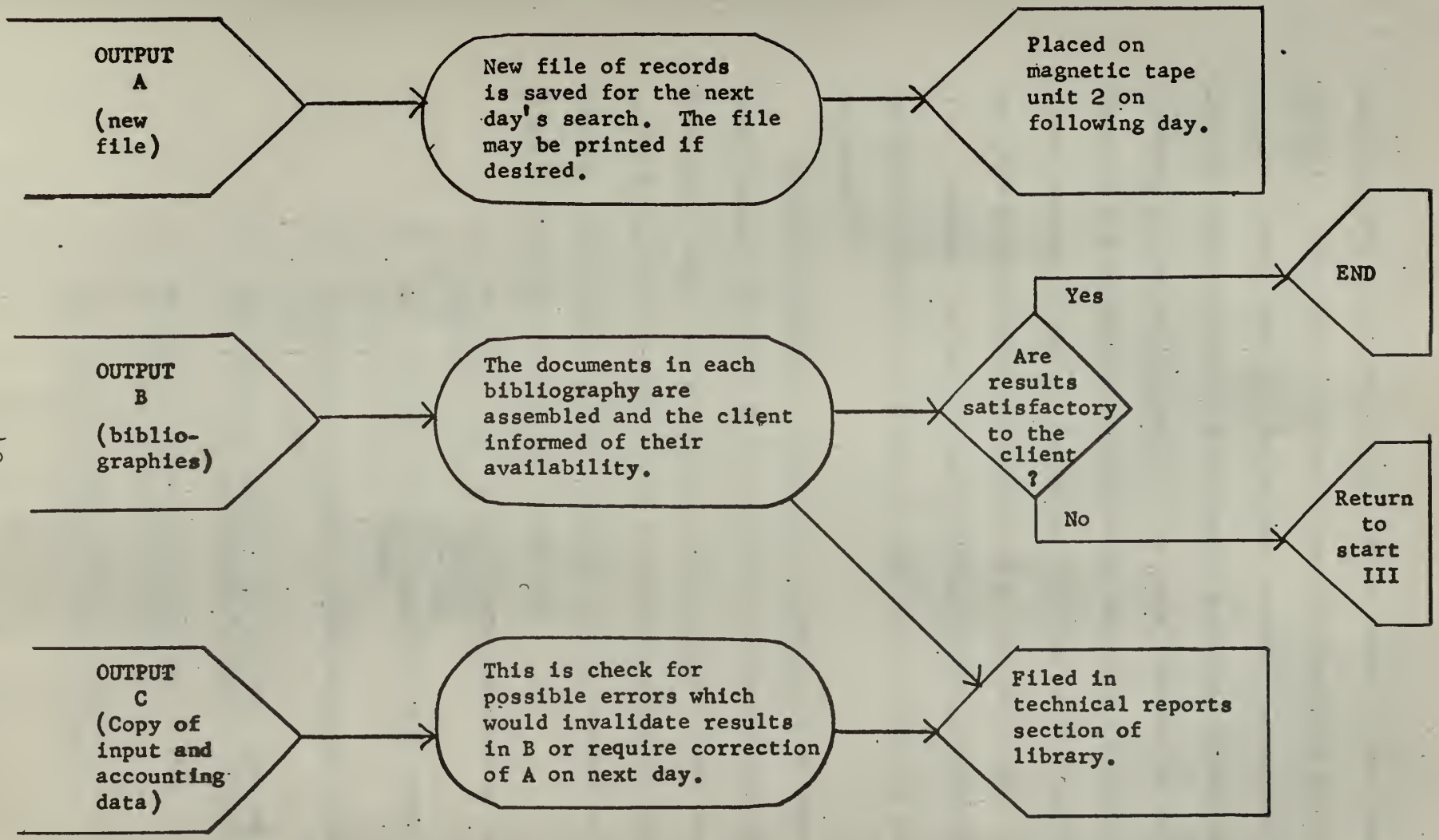


Figure 3.1C Operation Flow Chart of SABIRS

numbers headed by the identifying name or number of the request which they fill. This list is printed across the page with the tag name on the left. Every time this tag appears the shelf numbers to the right represent additions to the bibliography generated by that request. Following all the bibliographies, the input requests as read by the computer, are printed to facilitate the detection of errors and to permit the carrying along of clerical information (see Appendix C.). Next, are printed the new records just as they were added to the file. Thirdly, the list of shelf numbers which were deleted from the file is printed. Each line of this list is begun by the title "To delete".

The fourth block following the bibliographies is headed by the title "errors" and lists an identifying tag (the contents of the first word) every line of data which the program failed to print in its proper place in the output because of some technical error (parity, line length, incorrect format, etc.). Finally, a count is given of six items of interest in the run:

1. The number of blocks of item submitted to be deleted; titled "to delete".
2. The number of records submitted to be added to the file; titled "added".
3. The number of requests submitted to be filled; titled "requests"
4. The number of errors found (in the sense of the list in block four described above); titled "errors".
5. The actual number of records now on file at completion of the run; titled "records".
6. The number of items actually deleted from the old file during this run; titled "deleted".

The file of records itself may be printed on the line printer if desired. Examples of such outputs are also shown in Appendix A. The simplest method for making changes in a record on file is to list its shelf number for deletion and the correct record with same number as an addition to the file. Both can be accomplished at the same operating session. Note that the compiler does not maintain any order of shelf numbers on magnetic tape.

2. Error Analysis and Correction

The complier program may cause several error signals to be typed on the consol typewriter in the course of its operation.

"Unlisted"

This means that a character has appeared on the input tape which is not included in the dictionary of meaningful symbols. A space is substituted in the output. Operation is not halted. If the unlisted character appeared in a shelf number of an item to be deleted, the record will not have been deleted and should be repeated on the next day's run. If the unlisted character appeared in a record to be added, the record is now incorrectly stored in the file. Therefore it's number should be included as one to be deleted and the record should be repeated correctly among those to be added on the next day's run. If the unlisted character appeared in a request, the bibliography produced may contain items which are not pertinent. Hence, the request should be repeated correctly in the next day's run.

"Too Long"

This means that more than 120 characters, exclusive of spaces, have appeared in the input without a double period mark. This particular typographical error is singled out because it may cause invalidation of two or more requests or new records. However, the operation is not halted. The error and its consequences will be immediately obvious from the printed output. Corrective measures are the same as those for the "unlisted" signal.

"Not Even"

This means that the number of characters between two pairs of double period marks is not an even multiple of eight, as it should be if correct code words only are employed. Operation is not halted. This may not indicate an error if it occurs in a request where additional clerical information has been added after the coding (see Appendix C). In all other cases, a typographical error is indicated which may invalidate the results. The same remarks made in the case of the "unlisted" signal pertain here.

"Mode"

This means that the paper tape reader is not in the character mode. Operation of the program halts at address 06065. Clear computer, restart paper tape in character mode, and begin again.

"Taperror"

This signal appears in the event of any of several errors occuring in the reading or writing of magnetic tape. It indicates that a line of output data has been dropped. An identifying tag for the

line dropped appears in the list of "errors" in the output. (See Section 1.)

"No Date"

This error signal appears not on the typewriter, but on the magnetic tape output. It is printed in the bibliographic list compiled in response to any request in which no specific range of dates of publication is given. This usually does not indicate an actual error, but it is noted because a typographical error of almost any kind in a request which does specify date will cause the computer program to ignore the date specification. On the other hand, the computer will be unaware of mistakes in the other fields except for those noted by the previous error signals.

3. Special Capabilities and Restrictions

SABIRS, as actually programmed, allows for considerable variation in format and procedure over and above the routine operation already described. Some of these capabilities and the rules for using them are described in this section.

Direct File Copying

It is possible to copy from one magnetic tape reel to another a complete or partial file of records without change and without using any paper tape input. Place the file of records to be copied on tape unit two and the blank reel on tape unit three. Set selective jump key two. Start with the program address register equal to 07000. After pertinent error signal is "taperror" which has the same meanings as given in Section 2 of this Chapter.

Format of a Record

The format of a record of a document is precisely fixed, as the following example shows:

U00534740010000600005912000004460000167500001676..

Each eight characters is a separate code. Reading from left to right, the first eight characters is the shelf number; the second eight characters is the code for the source of the document; the third eight characters is the date of publication; and the last three sets of eight characters each represents a uniterm or descriptor associated with the document. The double period indicates the end of the record. Thus, in the case above:

U0053474	is the shelf number (the "U" indicates the unclassified area of the stacks).
00100006	is the source code.
00005912	is the date of publication (December, 1959).
00000446	is the code for a uniterm.
00001675	is another uniterm.
00001676	is another uniterm associated with this document.
..	marks the end of the record (there may have been as many as nine more uniterms).

The order, shelf number, source code, date, uniterms-- is fixed. Furthermore, each code consists of exactly eight characters, and zeros to the left must be typed (punched). None of the fields may be omitted. Up to 12 uniterms may be included, for a maximum of 120 characters to a record. Each record must end with a double period

which is not included in the character count. Spaces may be used anywhere in a record for ease in reading the typed copy; they are ignored by the computer and are not included in the character count.

Format of A Request

On the other hand, the format of a request allows for considerable variation, as may be seen from the numerous examples in Appendix C. The first eight characters are taken as a tag which identifies the request throughout the system's operations. After this tag, codes representing sources and uniterms may be mingled in any order as long as each one consists of eight characters. The coding for date in a request may also be placed anywhere among the other codes, but it consists of 16 characters in the fixed form: datexxxxthruyyyy, where the 'x's and y's represent two year-month date codes as in the record. Conventionally, one codes "since March 1960" as "date6003thru9999" and "before March 1960" as "date0000thru6003". After all codes are entered, the request may be filled out to a maximum of 120 characters (not including spaces which are ignored) with any clerical or other information desired. Every request must end with a double period.

Other Restrictions In Format

A list of items to be deleted consists of their shelf numbers (each eight characters) typed (punched) successively following the title "to delete", a double period after each set of up to 14 items.

The input punched paper tape may contain a date in the form: 06/08/61 (June 8, 1961) followed by a double period. The deletions,

additions, and requests may be listed in order. In one run, up to 1000 items may be deleted, 5000 new records may be added and 64 requests filled at the same time.

Use of Multiple Input Tapes

The input may consist of many physically separate pieces of paper tape. The breaks in input may occur anywhere, but each piece of tape must end with "seventh level" punch. This stops the program until another tape is loaded in the reader and the compiling is resumed by raising the start switch. When the last piece of tape is loaded and before starting the computer, selective jump switch 1 should be raised.

Use of Multiple File Tapes

When the file of records is of such size as to require the use of more than one real of magnetic tape for its storage, the following procedure should be used:

(a) Place completely filled reel of tape on unit two and blank reel on unit three.

(b) Do not set jump key two. Start program as usual, feeding in one or more paper tapes of input.

(c) Program will stop and wait for magnetic tape reel to be changed.

(d.1) If no updating is taking place, replace file tape on unit two with another reel of file, restart program from current pause.

(d.2) If updating of the file is being accomplished, remove and replace both unit two and three reels of magnetic tape. Put another file reel on unit two and a blank on three as usual. Restart from current pause condition.

(e) Whenever the last reel of file is about to be processed, set jump key two before restarting.

BIBLIOGRAPHY

1. Opler, Ascher and Baird, Norma. "Experience in developing information retrieval systems, on large electronic computers", in International Conference on Scientific Information, Washington Nov. 16-21, 1958. Vol. 21, Washington, 1959. p. 699-710.

2. Symposium on Information Storage and Retrieval Theory Systems, and Devices, Washington, D.C., 1958. Edited by Mortimer Taube and Harold Wooster. Columbia Univ. Press, New York, 1958. 228p.

3. Mooers, Calvin N. "A mathematical theory of language symbols in retrieval", in International Conference on Scientific Information, Washington, D.C., Nov. 16-21, 1958. Vol. 2, Washington, D.C. 1959. p. 1327-1364.

4. Mann, Margaret. Introduction to cataloging and the classification of books. Chicago, American Library Association, 1943. 276p.

5. Vickery, B. C. "The structure of information retrieval systems" in International Conference on Scientific Information, Wash. D.C., Nov. 16-21, 1958. Vol. 2, Washington, 1959. p. 1275-1290.

6. Barden, William A. and others. Automation of ASTIA, a preliminary report, December, 1959. AD-227-000. Armed Services Technical Information Agency, Arlington 12, Virginia. 50p. illus. Unclassified.

7. Shera, J. H. and others. Information systems in documentation. New York, Interscience, 1957. Vol. 2. 631p.

8. Cherenin, V. P. "The basic types of information tasks and some methods of their solution", in International Conference on Scientific Information, Washington, D.C., Nov. 16-21, 1958. Vol. 2, Washington 1959. p. 823-854.

9. Perry, J. W. and Kent, Allen Tools for machine literature searching...New York, Interscience, 1958. 972p. Tables.

10. Fano, R. M. "Information theory and the retrieval of recorded information." in Documentation in Action, by Jesse H. Shera and others. New York, Reinhold, 1956. p. 238-244.

11. Wise, Carl S. "Mathematical analysis of coding systems", in Casey, Robert S., and others, ed: Punched card: their applications to science and industry, second edition. New York Reinhold, c1958. p. 438-464

12. Ledley, Robert Steven. "Codifying: error correction and superimposition," in Digital computer and control engineering.... New York, McGraw-Hill, 1960. p. 242-255.

13. Grabbe, Eugene M., ed. Handbook of automation, computation, and control, Vol. 2; Computers and data processing...Los Angeles, Thompson Ramo Woolridge Inc. New York, John Wiley, c1959. illus.

14. Maloney, Clifford J. "Abstract theory of retrieval coding", in International Conference on Scientific Information, Washington, D.C. Nov. 16-21, 1958. Vol. 2. Washington, 1959 p. 1365-1382.

15. Estrin, Gerald. "Maze structure and information retrieval," in International Conference on Scientific Information, Washington, Nov. 16-21, 1958. Vol. 2, Washington, 1959. p. 1383-1414.

16. Selfridge, Oliver G., and Neisser, Ulric. "Pattern recognition by machine," reprinted from Scientific American, August 1960. 10p.

17. Luhn, H. P. "Identification of geometric patterns by topological description of their envelopes," in IBM Technical report, code: 00.011.599, 23 April, 1956.

18. Moore, Edward F. "The shortest path through a maze," in Proceedings of an International Symposium on the theory of Switching, parts I and II. Harvard Univ. Press, 1959.

19. Perry, J. W.and Kent, Allen. Documentation and information retrieval, an introduction to basic principles and cost analysis...Cleveland, Ohio, Western Reserve Univ. c1957. 156p.

20. Birkhoff, Garrett. "What is a lattice?" in The American Mathematical Monthly, Vol. 50, No. 8, October 1943, p. 484-486.

21. Birkhoff, Garrett. Lattice theory...revised edition. New York, American Mathematical Society, 1948. 283p. American Mathematical Society Colloquium Pub. Vol. XXV.

22. Birkhoff, Garrett and MacLane, Saunders. Algebra of classes, chapter XI, in A Survey of Modern Algebra...New York, The Macmillan Company, 1946. p. 311-332.

23. Jacobson, Nathan. Lattices, chapter VI, in Lectures in abstract algebra, Vol. 1, basic concepts. New York, van Nostrand, 1951. p. 187-211.

24. Hermes, Hans. Einfuhrung in die verbandstheorie...Berlin, Springer-Verlag, 1955. 164p. illus.

25. International Conference on Scientific Information, Washington, Nov. 16-21, 1958. Vol. 2, Washington, National Academy of Sciences, 1959.

26. Tannenbaum, Robert. "Overcoming barriers to the acceptance of new ideas and methods," in Electronic Data Processing for Business and Industry, by Richard G. Canning, New York, Wiley, 1956.

27. Becker, Esther R. and Murphy, Eugene F. The office in transition, meeting the problems of automation...New York, Harper, 1957. 190p.

28. Control Data Corporation. Characteristics of the Model 1604 computer, publication No. 018a. 1959.

29. Control Data Corporation. 1607 Magnetic Tape System Instruction Book, Vol. 1: Description and Operation, publication No. 037.

30. International Business Machines. 704 data processing system, reference manual, c1958. New York. 100p. illus.

31. International Business Machines Corp. Principles of operation; IBM accounting machine 402, 403, and 419. Revised January, 1956. 157p. Tables and charts.

32. Flores, Ivan. Computer logic; the functional design of digital computers. Englewood Cliffs, N.J., Prentice-Hall, 1960. 458p. illus.

33. Remington Rand Univac. Directions in the retrieval of scientific information, AD 228 389, Armed Services Technical Information Agency, Arlington 12, Virginia. 30p. Unclassified.

34. Gotlieb, C. C. and Hume, J.N.P. High-speed data processing... New York, McGraw-Hill, 1958. 338p.

APPENDIX A

SURVEY OF LATTICE THEORY[1]

We begin with an undefined entity, S, a set (or collection) of elements or members, a,b,c,...., whose nature is immaterial but which may well be sets themselves.

1. Partial Order

Definition I: The set A is a subset of the set B if and only if each member of A is also a member of B. A is called a proper subset of B if (1) it is a subset of B and (2) there exists a member of B which is not a member of A.

Definition II: A partially ordered set is a system consisting of a set S and a relation $\geq$ ("greater than or equals" or "contains") satisfying the following postualtes:

P_1 For all x, $x \geq x$ (Reflexivity)

P_2 If $x \geq y$ and $y \geq x$, then $x=y$ (Anti-symmetry)

P_3 If $x \geq v$ and $v \geq z$, then $x \geq z$ (Transitivity)

If x and y are any elements of S, we may have $x \geq y$ or $y \geq x$ or neither.

Definition III: If every pair of elements of a partially ordered set S are comparable (either $x \leq y$ or $y \leq x$), then S is said to be linearly ordered or to be a chain.

1. The material in this appendix is taken chiefly from comprehensive works by Birkoff [20,21], Birkoff and MacLane [22], Jacobson [23] and Hermes [24]. Since each theorem and definition appears in at least three of these references, individual citations have not been made.

If $x \geq y$ and $x \neq y$, then we will write $x > y$, and we also agree to use $y \leq x$ and $y < x$ as alternatives for $x \geq y$ and $x > y$.

Definition IV: In a finite partially ordered set, we say that x is a cover of y if $x > y$ and no u exists such that $x > u > y$.

It is clear that, if $x > y$ in a finite partially ordered set, then we can find a chain

$$x = u_1 > u_2 > \ldots > u_n = y$$

in which each u_i covers u_{i+1}. Conversely the existence of such a chain implies that $x > y$.

2. Lattices

Definition V: An element u of a partially ordered set S is said to be an upper bound for the subset A of S if $u \geq a$ for every a in A. If u is an upper bound and $u \leq v$ for any upper bound v of A, then u is a least upper bound (l.u.b.) of A.

Definition VI: An element u of a partially ordered set S is said to be a lower bound for the subset A of S if $u \leq a$ for every a in A. If u is a lower bound and $u \geq v$ for any lower bound v of A then u is a greatest lower bound (g.l.b.) of A.

We denote the l.u.b. of x, y by $x + v$ ("x union y", "x or y") and the g.l.b. by xy ("x intersecty", "x and y").

Definition VII: A lattice is a partially ordered set in which any two elements have a least upper bound and a greatest lower bound.

Definition VIII: A lattice, L, is said to be closed if any (finite or infinite) subset $A = a_i$ has a l.u.b. $\sum a_i$ and a g.l.b. $\prod a_i$.

Given a partially ordered set S, we mean by the lattice closure of S, the smallest closed lattice, S^*, which contains S.

Theorem I: A set L is a lattice if and only if the following algebraic identities hold:

L_1 For all x, $xx = x+x = x$

L_2 $xy = yx$ and $x+y = y+x$

L_3 $x(yz) = (xy)z$ and $x+(y+z) = (x+y)+z$

L_4 $x(x+y) = x+xy = x$

Definition IX: A subset M of a lattice L is called a sublattice (of L) if it is closed relative to the compositions union and intersection.

It is evident that a sublattice is a lattice relative to the induced compositions. On the other hand, a subset of a lattice may be a lattice relative to the partial ordering $\geqslant$ defined in L without being a sublattice. For example, let G be a group, let B be the lattice of subsets of G, and L be the lattice of subgroups of G. Then it is clear that L is contained in B, and that $H_1 \geqslant H_2$ has the same significance in these two sets. On the other hand, if H_1 and H_2 are subgroups, then $H_1 + H_2$ as defined in B is the set

sum of these groups. In general, this is not a subgroup; hence, it differs from the $H_1 + H_2$ as defined in L as the smallest subgroup of G containing their set sum.

3. Types of Lattices

Definition X: A lattice is called <u>modular</u> if it satisfies the condition

L_5 If $x \geq y$, then $x(y + z) = y + xz$

Definition XI: A lattice is called <u>strongly distributive</u> if it satisfies the condition

L_6 $x(y + z) = xy + xz$

Theorem II: If three elements in a lattice satisfy L_6, then they satisfy its dual

L_{6d} $(x + y)(x + z) = x + yz$

Theorem III: In all finite lattices, there exist elements O and I which are universal lower and upper bounds respectively; that is, for all x, $O \leq x \leq I$.

Definition XII: A lattice is called <u>complemented</u> if it satisfies the condition that for every x in L there exists an x' such that

L_7 $xx' = O$, $x + x' = I$

Definition XIII: A lattice is called <u>Boolean</u> if it is both strongly distributive and complemented.

4. Chain Conditions

Let a and b be two elements of a modular lattice such that $a \geq b$. We consider now the finite descending chains

$$a = a_1 \geqslant a_2 \geqslant \cdots \geqslant a_n = b$$

connecting a and b.

Definition XIV: One such chain is called a refinement of a second if its terms include all the terms of the other chain. Two chains are said to be equivalent if their terms can be put in one-to-one correspondence such that corresponding intervals $I(a_i, a_{i+1})$ of the two chains are isomorphic.

Theorem IV: Any two finite descending chains connecting the elements a,b $(a \geqslant b)$ of a modular lattice have equivalent refinements.

Definition XV: A composition chain connecting a,b, $a > b$ is a finite sequence

$$a = a_1 > a_2 > a_3 > \cdots > a_n = b$$

in which each a_i is a cover of a_{i+1}.

We assume for simplicity now that L contains 0 and I, and we take a = I, b = 0 in the foregoing.

Definition XVI: If there exists a composition chain in L, connecting I and 0, L is said to be of finite length. The number of intervals of this chain is called the length (dimension) of L.

Theorem V: A modular lattice with 0 and I is of finite length if and only if the following two chain conditions hold:

Descending chain condition. There exists no infinite properly descending chain, $a > b > c > \cdots$ in L.

Ascending chain condition. There exists no infinite properly ascending chain, $a < b < c < \ldots$ in L.

Definition XVIII: We say that an element a in L is (intersection or meet) reducible if $a = b_1 b_2$ where the $b_i > a$.

Definition XVIII: We say that the representation of a as the intersection of m elements is irredundant if the intersection of any m-1 of them contains a properly.

Theorem VI: The number of terms in any two irredundant representations of an element as g.l.b. of irreducible elements is the same.

Definite XIX: An element p of a lattice with 0 is called a point if p is a cover of 0.

Theorem VII: If L satisfies the descending chain condition, L contains points.

Theorem VIII: If L is a complemented modular lattice that satisfies both chain conditions, then the element I of L is a l.u.b. of independent points. Conversely, if L is a modular lattice with 0 and I such that I is a l.u.b. of a finite number of points, then L is complemented and satisfies both chain conditions.

5. Cardinal Products

Definition XX: Let X and Y be sets, each with a partial ordering relation $\geq$. By the cardinal product XY, we mean the set of all pairs x,y (x in X, y in Y), where

$(x,y) \geq (u,v)$ means $x \geq u$ in X and $y \geq v$ in Y. X and Y are the components of the cardinal product.

Theorem IX: The cardinal product LM of two lattices L and M is also a lattice. Furthermore, LM will satisfy one of the previously stated conditions L_i if and only if both L and M satisfy L_i.

6. Homomorphisms of Lattices

We shall now consider many-to-one correspondences T: L M between lattices. The following three properties (among those which T may possess) are of interest.

(1) $x \geq y$ implies $T(x) \geq T(y)$

(2) $T(xy) = T(x)T(y)$

(3) $T(x + y) = T(x) + T(y)$

Note that T may possess all or none of these properties. Also, (2) implies (1); (3) implies (1); but (1) does not imply either (2) or (3).

Definition XXI: T is called isotone if it satisfies (1) but not (2) nor (3).

T is called a meet-homomorphism if it satisfies (2) but not (3).

T is called a join-homomorphism if it satisfies (3) but not (2).

T is called a lattice-homomorphism if it satisfies (1), (2), and (3).

Definition XXII: A subset J of elements of a lattice L is an ideal if and only if

x in J and y in J imply x + y in J, and

x in J and $v \leq x$ imply v in J.

As usual, a lattice-homomorphism of L can be used as an equivalence relation to partion L into congruence classes of elements.

Theorem X: The antecedents of any element under any lattice-homomorphism form a convex sublattice, or sublattice which contains with any a,b, all elements between a and b.

In complemented lattices, every congruence relation (hence every lattice-homomorphism) is determined by the ideal of elements congruent to O.

Theorem XI: Any complemented lattice of finite length is a cardinal product of "simple" complemented lattices.

Definition XXIII: The center of a partly ordered set P with O and I is the set of elements e in P which have one component I and the others O under some direct factorization of P as a cardinal product of other partially ordered sets.

Theorem XII: The center of a Boolean lattice L is the intersection of its maximal Boolean sublattices.

APPENDIX B

EXAMPLE OF SUPERIMPOSED CODING

The following example is intended to illustrate the principle of superimposed coding.

In a 20 bit computer word let three characteristics have the following codes:

Characteristic	Position																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1st.....	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0
2nd.....	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
3rd.....	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1

then the corresponding superimposed codification is

0 1 1 0 0 1 0 1 0 0 1 1 0 0 1 1 0 0 0 1

With this example we shall first illustrate how superimposed coding can be made to use fewer than $X \log_2 N$ bits and yet maintain the ability to distinguish among N characteristics. Suppose that $N = 4{,}500$. Let us code the characteristics as follows: use 20 bit positions, and let each code contain precisely 4 units. This system will allow us to code each characteristic uniquely, because 20 positions can be taken four at a time in

$$\binom{20}{4} = \frac{20!}{4!(20-4)!} = 4{,}845 \text{ different ways, and}$$

$4{,}845 > 4{,}500$. Finally, suppose that each item is associated with three characteristics; that is, $X = 3$. Then let the coding of the item be the superimposed codings of its associated three characteristics. Note that the coding of an item now requires only 20 bit

positions, rather than the 39 considered necessary for efficient coding; that is, $X \log_2 N = 3 \log_2 4500 = 39$.

Now it is clear that the item coded in the example above will be selected if all items having the first characteristic are searched for. Similarly for the second or third characteristic. However, we must evidently pay a price for this convenience and saving in item-coding positions. Suppose that the code for some other characteristic is

0010 00001 00001 00001 0000

Our item is not associated with this characteristic but it will be selected in a search for items associated with this characteristic. The reason for this is that our codifications overlapped, allowing for more than three combinations of four ones to give rise to the item coding. In the example, nine ones appear, allowing for

$$\binom{9}{4} = \frac{9!}{4!(9-4)!} = 126$$

such possibilities, and hence 126 - 3 = 123 possible false combinations. If not all the 4,845 possible characteristics actually appear, the situation is alleviated somewhat.

The use of superimposed coding therefore requires the determination of the probability p that an item not associated with a particular characteristic will be selected during the search for that characteristic. The formulas given below are taken from Ledley [12] and assume that the codes for characteristics and the association of items with characteristics are random.

Let H = number of positions in superimposed coding.

Y = number of ones in a characteristic code.

X = number of characteristics associated with an item.

Then
$$p = \frac{\left[\binom{H}{Y} - 1\right]^X - \sum_{i=0}^{i=Y-1} E_{Y-i}}{\binom{H}{Y}^X}$$

Where
$$E_Y = \binom{Y}{0}\binom{H-Y}{Y}^X$$

$$E_{Y-1} = \binom{Y}{1}\binom{H-Y+1}{Y}^X - \binom{Y}{1}E_Y$$

$$E_{Y-2} = \binom{Y}{2}\binom{H-Y+2}{Y}^X - \binom{Y}{2}E_Y - \binom{Y-1}{1}E_{Y-1}$$

$$E_{Y-3} = \binom{Y}{3}\binom{H-Y+3}{Y}^X - \binom{Y}{3}E_Y - \binom{Y-1}{2}E_{Y-1} - \binom{Y-2}{1}E_{Y-2}$$

and so forth .

For the example here, H 20, Y 4, X 3: hence

$$p = \frac{\left[\binom{20}{4} - 1\right]^3 - \left(E_1 + E_2 + E_3 + E_4\right)}{\binom{20}{4}^3} = 0.0377$$

APPENDIX C

EXAMPLES OF PRODUCTION DATA

This appendix contains examples of some typical data processed by SABIRS I, the first version of a semi-automatic bibliographic information retrieval system actually installed and operating on a limited basis at the Technical Reports Library of the U. S. Naval Postgraduate School.

Figure C.1 is a portion of a typical file of records as recorded on magnetic tape for use of the system. An actual file may contain up to 75,000 records on one reel of tape.

Figure C.2 is an example of input data; it contains the following:

(a) Four records to be added to the file. Note that two of these have the same shelf number, date, and source, but different uniterms. This demonstrates the handling of a record whose analysis gives rise to more than the twelve uniterms permitted in each record.

(b) Two lists of items to be deleted for a total of four deletions. Note the clerical information appearing here and after the requests. Any such notes may be entered after the coded data up to a total of 120 characters per block of information.

(c) Seven requests for bibliographic information. Note that the identifying tags used (in this case the names of the requestors) must be exactly eight characters long not counting spaces.

Figure C.3 is the information output from this input data. There are seven blocks of data separated by blank lines.

(a) Bibliographies requested. Note that each time a particular tag appears, that line consists of more documents to be included in the particular bibliography.

(b) Copy of the requests as received.

(c) Copy of the new records to be added, printed as received by the computer.

(d) Copy of the lists of items to be deleted, as received.

(e) List of errors. (See Chapter III, section 2, "Taperror").

(f) Accounting data and date.

Figure C.4 shows the same portion of a file as in Figure C.1 after updating by the additions and deletions appearing in Figures C.2 and C.3.

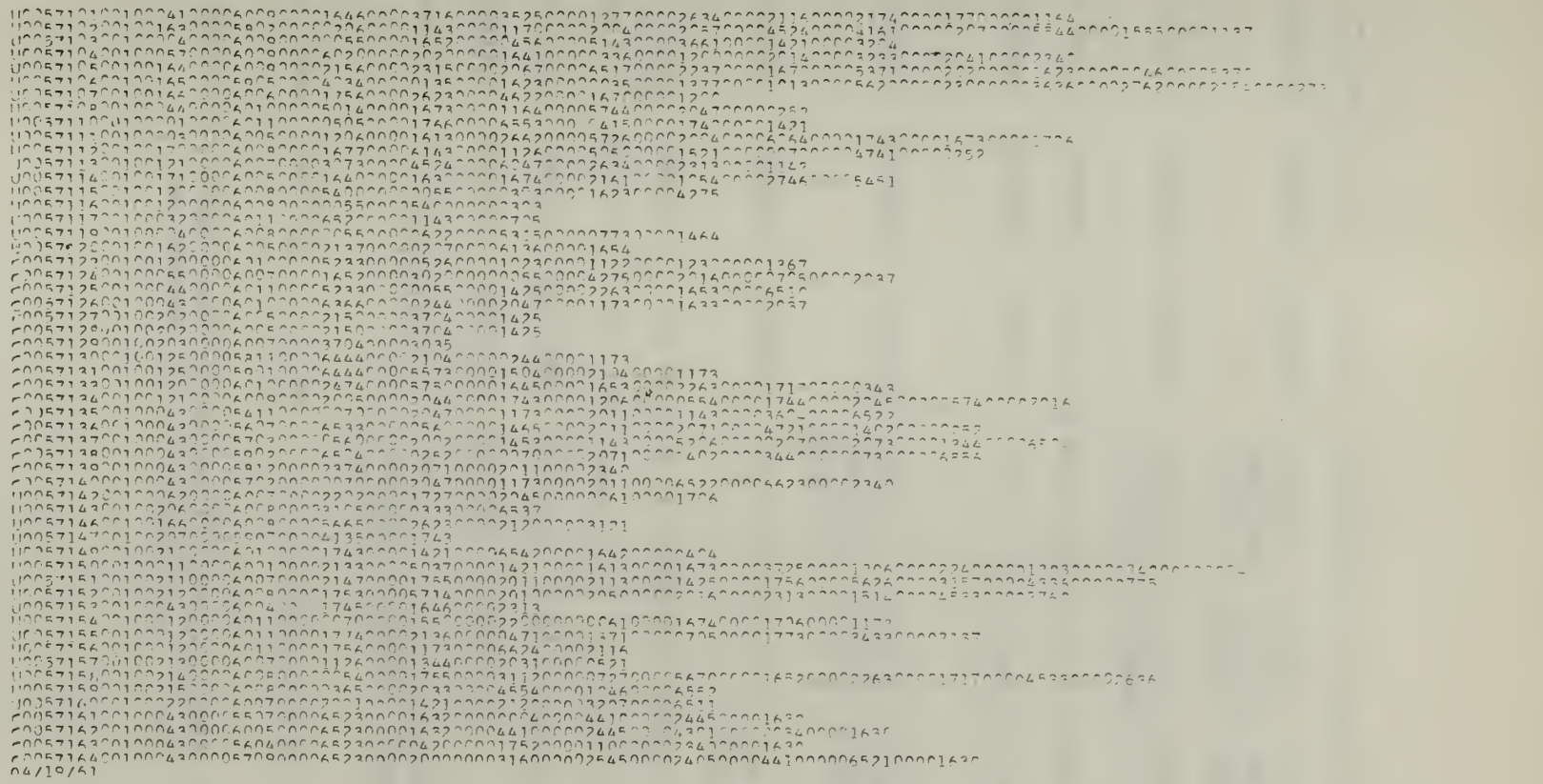


Figure C.1 Portion of File of Records

```
to delete u0057147  u0057106  superceded by final reports..

u0057036  00100126  00005911  00006444  00002104  00002017  00001715
                              00001640  00003403  00001615  00001503
                              00001670  00002116  00003157  00001623..

u0057036  00100126  00005911  00003150  00000774  00004040  00005730
                              00002306  00000340..

McCalla/  00000055  date 6006 thru 9999  for Lt. T.R. McCalla, box 1677..

Jauregui  00100043  00002047  for Lt. S. Jauregui, box 329..

05/19/61..

U0057109  00100167  00006005  00002021  00001727  00001706  00002004
                              00001743  00001203  00004324  00004710
                              00005457..

Abernath  00006444  00001653  date 5803 thru 6001 for Capt. T.R. Abernathy
                                                  box 1420..

Wildberg  00100012   for Lt. A.M. Wildberger, box 1439..

Henn////  00001717   date 5906 thru 9999  for Capt. H.R. Henn, box 1211..

to delete c0057205  c0057122  c0057129  destroyed per DOD DIR 5200.10..

u0057148  00100011  00006099  00000340  00001727..

Door////  00102064  00100732  00100037  date 6003 thru 9999  00006437
                              00003207  00000462  for W. T. Door phone
                              ext. 488..

Prof. Doe  00001727  for Prof. J.Q. Doe, code 018 D2..
```

Figure C.2 Sample Input Data Containing Deletions, Additional Records, and Requests for Information.

```
PROF.DOE        NO DATE     U0057109    U0057142    U0057079    U0057068    U0057038    C0057009
MCCALLA/                    C0057125    C0057124    U0057119    U0057116    U0057115    U0057103
JAUREGUI        NO DATE                                         C0057140    C0057135    C0057126
ABERNATH
WILDBERG        NO DATE                             U0057156    U0057155    U0057154    C0057008
HENN////                                            U0057158    C0057133    U0057080    U0057062
DOOR////
PROF.DOE                                                                                U0057148

MCCALLA/00000055DATE6006THRU9999FORLT.T.R.MCCALLA,BOX1677
JAUREGUI00100043000020475ORLT.S.JAUREGUI,BOX329
ABERNATHC0006444000016530ATE5803THRU6001FORCAPT.T.R.ABERNATHY,BOX1420
WILDBERG00100012FORLT.A.M.WILDBERGER,BOX1439
HENN////00001717DATE5906THRS9999FORCAPT.H.R.HENN,BOX1211
DOOR////001020640010073200100037DATE6003THRU999900006437000032070000046 2FORW.T.DOOR,PHONEEXT.488
PROF.DOE0 001727FORPROF.J.O.DOE,CODE01802

U0C57036001001260000591100006444000021040000201700001715000016400000340300001615000015030000167000002116000031570000162 3
U0057036001001260000591100003150000007740000404000005730000023060000034 0
U0057109C01001670000600500000202100001727000017060000200400001743000012030000432400004710000054 57
U0057148001000110000600900000340000001727

TODELETEU0057147U0057106SUPERCEDEDBYFINALREPORTS
TODELETEC0057205C0057122C0057129DESTROYEDPERDODDIR5200.10

ERRORS

00000002        TODELETE
00000004        ADDED
00000007        REQUESTS
0000000C        ERRORS
00000133        RECORDS
00000004        DELETED
05/19/61
```

Figure C.3 Output Resulting from Input Shown in Figure C.2

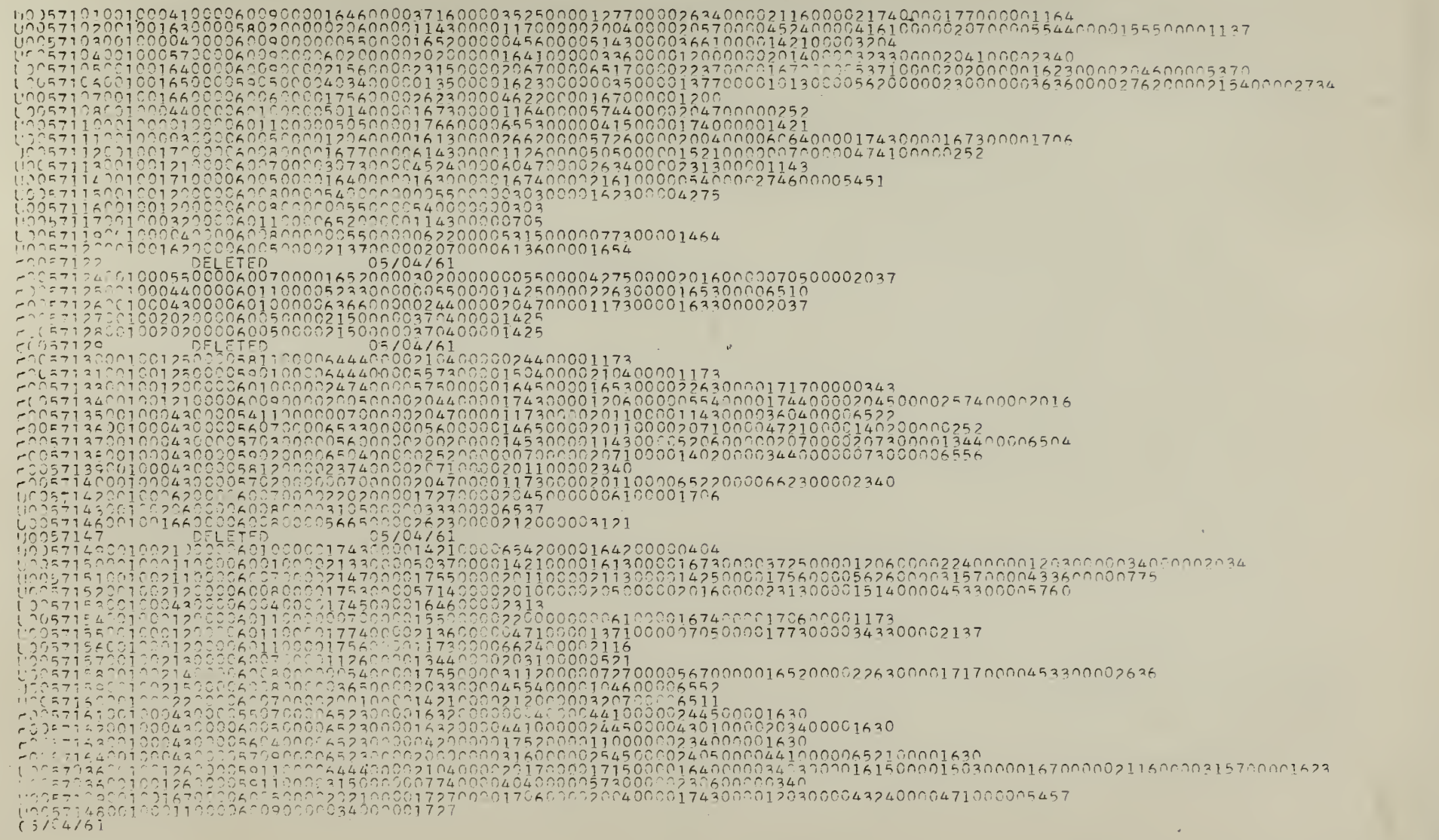


Figure C.4 Portion of File of Records After Updating